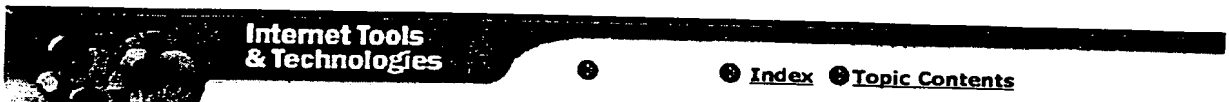


Index Topic Contents

Previous Topic: Application Desktop Toolbars
Next Topic: Briefcase Reconcilers

Band Objects

Microsoft® Internet Explorer 4.0 allows an application to define and create bands that are displayed within the shell and the browser.

- Band Object Types
- Implementing a Band Object

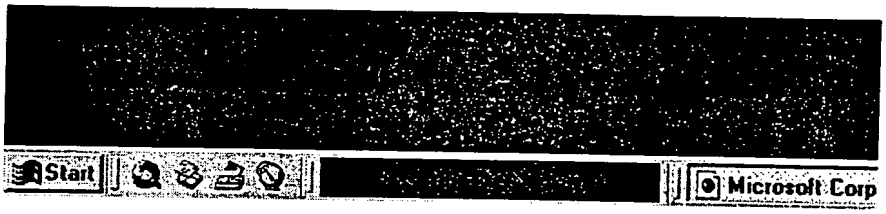
Band Object Types

Internet Explorer 4.0 currently supports three different types of band objects: Desk Bands, Explorer Bands, and Communication Bands. Each type of band serves a different purpose. The following sections describe these different band types and their capabilities.

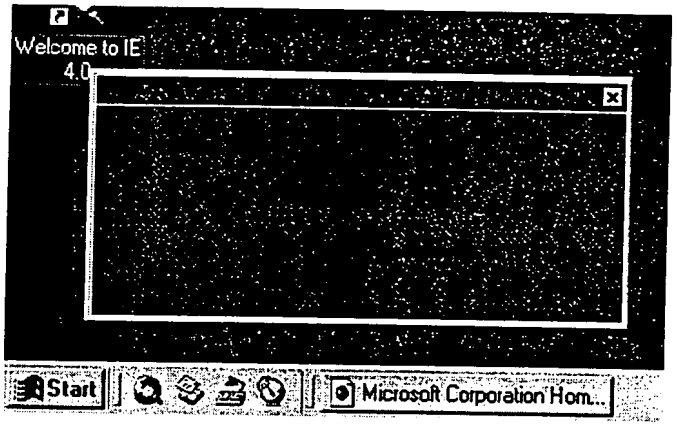
Desk Bands

Desk Bands are band objects that are displayed as a band in the shell taskbar or as a floating tool window on the desktop. Desk Bands are used to display information and to receive input from the user. They are available to the user throughout the entire Microsoft® Windows® session. Desk Bands are supported in shell versions 4.71 and later.

The following illustration shows a Desk Band on the taskbar.



This illustration shows a Desk Band as a floating toolbar.

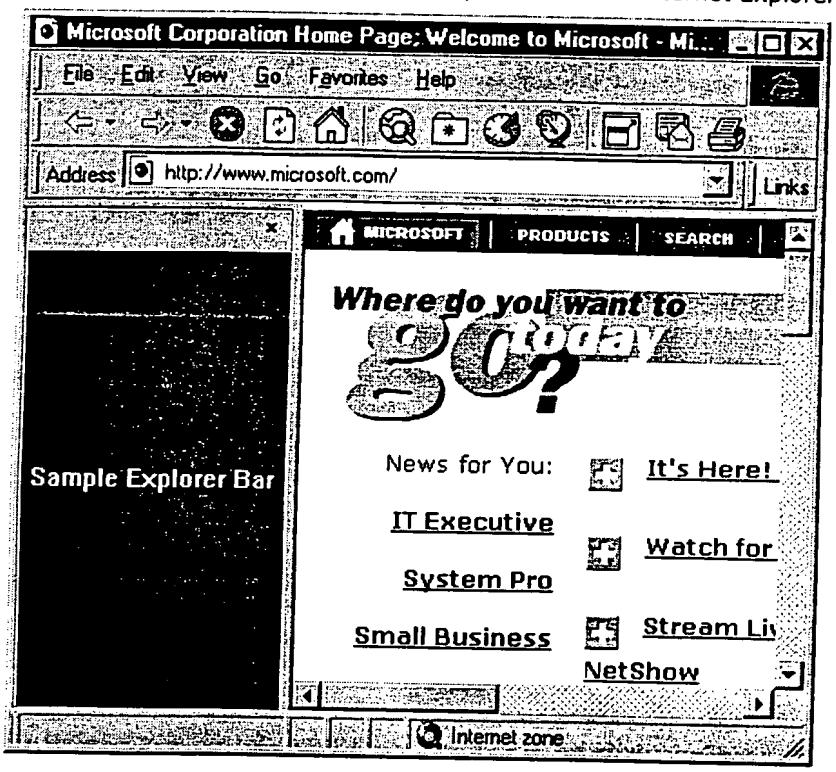


Explorer Bands

Explorer Bands are band objects that are displayed in a browser window. They usually contain

information or supply tools that are helpful to the user while using the browser. These bands are displayed to the left of the current view and normally receive user input that results in some action being performed. Explorer Bands are supported in browser versions 4.71 and later.

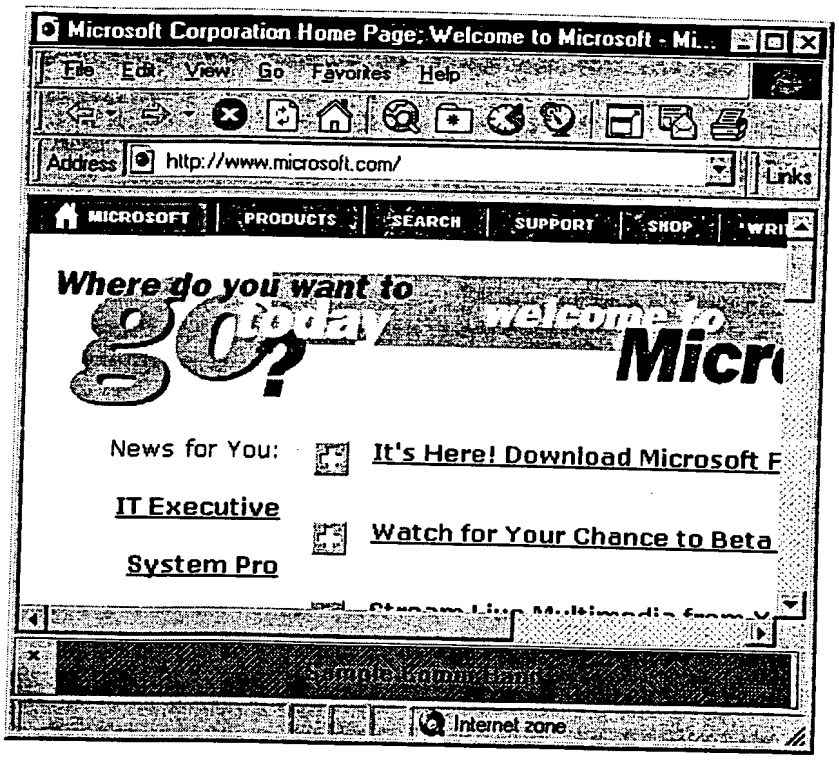
The following illustration shows an Explorer Band in Internet Explorer 4.0.



Communication Bands

Communication Bands are band objects that are displayed in a browser window. This can be a Windows Explorer window or an Internet Explorer 4.0 browser window. Communication Bands are normally used to relay information to the user. These bands are displayed below the current view and don't usually accept user input. Communication Bands are supported in browser versions 4.71 and later.

The following illustration shows a Communication Band in Internet Explorer 4.0.



Implementing a Band Object

This section describes the basic steps involved in creating a band object. To receive the most benefit from this information, you should be familiar with OLE COM, C++ programming, component categories, and the registry.

Registering the Band Object

The band object server must be registered as an OLE in-process server that supports the apartment threading model. The default value for the server is a text entry that will be displayed in the band selection menu. The following example shows the required registry entries.

```
HKEY_CLASSES_ROOT
    CLSID
        <server clsid> = "&Sample Band"
            InProcServer32 = "<server DLL path and file name>'
                             "ThreadingModel"="Apartment"
```

In addition to these entries, the band object server is registered for one or more component categories. The registered component categories define the type of band object that the server implements. The following list shows the available band object component categories. These component category identifiers are defined in Shlobj.h.

CATID_DeskBand The band object implements a Desk Band.

CATID_InfoBand The band object implements an Explorer Band.

CATID_CommBand The band object implements a Communication Band.

Band Object Interfaces

The following interfaces must be implemented by a band object:

- IDeskBand
- IObjectWithSite
- IPersist
- IPersistStream

If the band can accept user input, IInputObject is also required. And if you want to add to the context menu that is displayed for the band, you must implement the IContextMenu interface.

Implementing the IDeskBand interface in a band object

A band object must implement the IDeskBand interface. Because **IDeskBand** is derived from IDockingWindow, the band object must implement all the methods for both of these interfaces.

IDockingWindow is derived from IOleWindow. The **IOleWindow** interface provides the **GetWindow** method so that the container can obtain the band object's window handle.

The IDockingWindow methods are used to show, hide, and close the band object window. Unlike a normal docking window implementation, a band object does not have to negotiate its border space within the band container. Instead, the band container resizes the window provided by the band object's IOleWindow::GetWindow method.

The band container calls the IDeskBand::GetBandInfo method to obtain information about the band. In response to this method, the band object must provide all of the display band information that is being requested.

Implementing the IObjectWithSite interface in a band object

A band object must also implement IObjectWithSite. The band container uses this interface to set the band object's site with the **SetSite** method. The band object should perform the following steps when its **SetSite** method is called:

- Release any site pointer that is being maintained.
- If the site pointer passed to this method is NULL, the band is being removed. The band object can just return S_OK.
- If the site pointer passed to this method is not NULL, a new site is being set. In this case, the following steps should be performed:
  - Call QueryInterface on the site interface for IOleWindow, and call the IOleWindow::GetWindow method to obtain the parent window of the band object window. The **IOleWindow** interface must be released if it is no longer needed.
  - Create the band object window as a child of the parent window obtained in the previous step. The window should not be created as a visible window.
  - Call QueryInterface on the site interface for IInputObjectSite, and maintain this as the band object's site pointer. This step is necessary only if the band object implements IInputObject.
  - If all the previous steps succeeded, the band object should return S_OK. If any of these steps failed, the band object should return an OLE-defined error code indicating what failed.

The following example shows a very simple implementation of a **SetSite** method.

```
STDMETHODIMP CBandObject::SetSite(IUnknown* punkSite)
{
//If a site is being held, release it.
if(m_pSite)
   {
   m_pSite->Release();
   m_pSite = NULL;
   }

//If punkSite is not NULL, a new site is being set.
if(punkSite)
   {
   //Get the parent window.
   IOleWindow  *pOleWindow;

   m_hwndParent = NULL;

   if(SUCCEEDED(punkSite->QueryInterface(IID_IOleWindow, (LPVC
      {
      pOleWindow->GetWindow(&m_hwndParent);
      pOleWindow->Release();
      }

   if(!m_hwndParent)
      return E_FAIL;

   if(!CreateBandWindow(m_hwndParent))
      return E_FAIL;

   //Get and keep the IInputObjectSite pointer.
   if(SUCCEEDED(punkSite->QueryInterface(IID_IInputObjectSite,
      {
      return S_OK;
      }

   return E_FAIL;
   }

return S_OK;
}
```

Implementing the IPersist and IPersistStream interfaces in a band object

The band container uses the IPersist and IPersistStream interfaces to obtain information from the band object and to allow the band object to load and/or store any persistent information that it

needs.

IPersist has only one method—**GetClassID**—which the container uses to obtain the band object's class identifier.

The band container calls the band object's IPersistStream methods to allow the band object to save or load any persistent data that it requires. If the band object does not have any persistent data, it still must return success codes from the **IPersistStream** methods, even if no actions are performed in these methods.

Implementing the IInputObject interface in a band object

If a band object accepts user input, it must implement the IInputObject interface. The container uses **IInputObject** and implements IInputObjectSite to maintain proper user input focus when more than one contained window exists. For implementation and usage information, see the descriptions for the **IInputObject** and **IInputObjectSite** interfaces.

Implementing the IContextMenu interface in a band object

If a band object needs to add items to the context menu that is displayed for the band, it must implement the IContextMenu interface. The container will obtain the **IContextMenu** interface pointer by calling the band object's QueryInterface method, requesting IID_IContextMenu. The container uses the **IContextMenu** interface to obtain the band-specific context menu items and to execute the selected commands. For implementation information, see the descriptions for the **IContextMenu** interface.

Band Object Site Commands

The band object container implements an IOleCommandTarget interface that can be used to send commands to the container. This interface is obtained by calling the IInputObjectSite's QueryInterface method, requesting IID_IOleCommandTarget. The IOleCommandTarget::Exec method can then be called to send commands to the container. The command group is identified as CGID_DeskBand, and the following commands are supported:

DBID_BANDINFOCHANGED The band's information has changed. The container will call the band's IDeskBand::GetBandInfo method, requesting the updated information. The *pvaIn* argument of the IOleCommandTarget::Exec method is a VARIANT that contains the band identifier that was passed in the most recent call to **IDeskBand::GetBandInfo**.

DBID_MAXIMIZEBAND The container will maximize the band. The *pvaIn* argument of the IOleCommandTarget::Exec method is a VARIANT that contains the band identifier that was passed in the most recent call to IDeskBand::GetBandInfo.

Top of Page